CLAIMS

What is claimed is:

1. A processor, comprising:

a Boolean logic unit, wherein the Boolean logic unit is operable for performing the short-circuit evaluation of Normal Form Boolean expressions/operations;

a plurality of input/output interfaces, wherein the plurality of input/output interfaces are operable for receiving a plurality of compiled Boolean expressions/operations and transmitting a plurality of compiled results; and

a plurality of registers.

2. The processor of Claim 1, wherein the Boolean logic unit is operable for performing the short-circuit evaluation of Disjunctive Normal Form Boolean expressions/operations.

3. The processor of Claim 2, further comprising a p-bit OR register, wherein the p-bit OR register is operable for rolling up the results of disjuncts.

4. The processor of Claim 3, wherein the default value of the p-bit OR register is 'zero' or its logical equivalent.

5. The processor of Claim 3, wherein the p-bit OR register initializes to a value of 'zero', or its logical equivalent, after the start of a predetermined operational code.

6. The processor of Claim 3, wherein the p-bit OR register remains at a value of 'zero', or its logical equivalent, if all of the disjuncts of a Boolean expression/operation being evaluated are false.

7. The processor of Claim 3, wherein a Disjunctive Normal Form Boolean expression/operation is true if the p-bit OR register is set to 'one', or its logical equivalent, and the remainder of the Boolean expression/operation is short-circuited.

8. The processor of Claim 2, further comprising a q-bit AND register, wherein the q-bit AND register is operable for storing the results of the evaluation of one or more terms within disjuncts.

9. The processor of Claim 8, wherein the q-bit AND register initializes to a value of 'one', or its logical equivalent, and remains in that state until a state in a predetermined disjunct evaluates to 'zero' or its logical equivalent.

10. The processor of Claim 9, further comprising an r-bit AND disjunct register, wherein the r-bit AND disjunct register indicates that the evaluation of a disjunct comprising an AND clause has begun.

11. The processor of Claim 10, wherein the r-bit AND disjunct register initializes to a value of 'zero', or its logical equivalent, and remains in that state until an AND expression/operation sets its value to 'one' or its logical equivalent.

12. The processor of Claim 11, wherein the value of the r-bit AND disjunct register remains at 'one', or its logical equivalent, until reset by a Boolean OR operational code, an end of operation operational code, or a start of operation operational code.

13. The processor of Claim 10, wherein a predetermined disjunct evaluates to false if the q-bit AND register is set to 'zero', or its logical equivalent, and the r-bit AND disjunct register is set to 'one', or its logical equivalent, and the processor short-circuits to the start of the next disjunct.

14. The processor of Claim 10, wherein the OR register, the AND register and the AND disjunct register are each 1 bit wide.

15. The processor of Claim 10, wherein the OR register, the AND register and the AND disjunct register are each more than one bit wide.

16. The processor of Claim 1, further comprising an operation decoder, wherein the operation decoder is operable for deciphering an operational code and controlling units that are dependent upon the operational code.

17. The processor of Claim 16, wherein functions of the operation decoder comprise Boolean OR, Boolean AND, end of operation, no operation, unconditional jump, conditional jump, start of operation, and start of disjunct.

18. The processor of Claim 17, wherein the Boolean OR and Boolean AND functions are operable to OR or AND, respectively, the result of a comparison between a value from a memory location with an immediate value provided as part of an instruction.

19. The processor of Claim 17, wherein the Boolean OR and Boolean AND functions are operable to OR or AND, respectively, the result of a comparison between a first value from a first memory location with a second value from a second memory location.

20. The processor of Claim 17, wherein the Boolean OR and Boolean AND functions are operable to OR or AND, respectively, the result of a comparison between a first value provided directly by a device with a second value provided directly by a device.

21. The processor of Claim 17, wherein functions of the operation decoder further comprise AND compare and OR compare, each operable for comparing the result of a comparison between a first value and a threshold second value.

22. The processor of Claim 21, wherein the AND compare and OR compare functions are operable for comparing the result of a comparison between a value stored in a memory with an immediate value provided as part of an instruction.

23. The processor of Claim 21, wherein the AND compare and OR compare functions are operable for comparing the result of a comparison between a first value stored in a memory with a second value stored in the memory.

24. The processor of Claim 21, wherein the AND compare and OR compare functions are operable for comparing the result of a comparison between a first value provided directly by a device with a second value provided directly by a device.

25. The processor of Claim 21, wherein the AND compare and OR compare are each operable for determining whether the first value is greater than the threshold second value.

26. The processor of Claim 21, wherein the AND compare and OR compare are each operable for determining whether the first value is greater than or equal to the threshold second value.

27. The processor of Claim 21, wherein the AND compare and OR compare are each operable for determining whether the first value is less than the threshold second value.

28. The processor of Claim 21, wherein the AND compare and OR compare are each operable for determining whether the first value is less than or equal to the threshold second value.

29. The processor of Claim 17, wherein functions of the operation decoder further comprise a load memory function.

30. The processor of Claim 29, wherein the load memory function is operable to directly load an immediate value, provided as part of an instruction, into a location in a memory.

31. The processor of Claim 29, wherein the load memory function is operable to directly load a value from a first location in a memory into a second location in the memory.

32. The processor of Claim 1, wherein the Boolean logic unit is operable for performing the short-circuit evaluation of both Conjunctive Normal Form Boolean expressions/operations and Disjunctive Normal Form Boolean expressions/ operations.

33. The processor of Claim 32, further comprising a p-bit AND/OR register, wherein the p-bit AND/OR register is operable for rolling up the results of conjuncts and/or disjuncts.

34. The processor of Claim 33, wherein the default value of the p-bit AND/OR register is 'one', or its logical equivalent, when evaluating Conjunctive Normal Form Boolean expressions/operations and 'zero', or its logical equivalent, when evaluating Disjunctive Normal Form Boolean expressions/operations.

35. The processor of Claim 33, wherein the p-bit AND/OR register initializes to a value of 'one', or its logical equivalent, after the start of a predetermined operational code when evaluating Conjunctive Normal Form Boolean expressions/operations and a value of 'zero', or its logical equivalent, when evaluating Disjunctive Normal Form Boolean expressions/operations.

36. The processor of Claim 33, wherein the p-bit AND/OR register remains at a value of 'one', or its logical equivalent, if all of the conjuncts of a Conjunctive Normal Form Boolean expression/operation being evaluated are true, and remains at a value of 'zero', or its logical equivalent, if all of the disjuncts of a Disjunctive Normal Form Boolean expression/operation being evaluated are false.

37. The processor of Claim 33, wherein a Conjunctive Normal Form Boolean expression/operation is false if the p-bit AND/OR register is set to 'zero', or its logical equivalent, and the remainder of the Boolean expression/operation is short-circuited.

38. The processor of Claim 33, wherein a Disjunctive Normal Form Boolean expression/operation is true if the p-bit AND/OR register is set to 'one', or its logical equivalent, and the remainder of the Boolean expression/operation is short-circuited.

39. The processor of Claim 32, further comprising a q-bit AND/OR register, wherein the q-bit AND/OR register is operable for storing the results of the evaluation of one or more terms within conjuncts and disjuncts.

40. The processor of Claim 39 further comprising an r-bit OR conjunct/AND disjunct register, wherein the r-bit OR conjunct/AND disjunct register indicates that the evaluation of a conjunct comprising an OR clause, or a disjunct comprising an AND clause, has begun.

41. The processor of Claim 40, wherein the r-bit OR conjunct/AND disjunct register initializes to a value of 'zero', or its logical equivalent, when evaluating Conjunctive Normal Form Boolean expressions/operations and remains in that state until an OR expression/operation sets its value to 'one', or its logical equivalent, and initializes to a value of 'one', or its logical equivalent, when evaluating Disjunctive Normal Form Boolean expressions/operations and remains in that state until an AND expression/operation sets its value to 'zero' or its logical equivalent.

42. The processor of Claim 40, wherein a predetermined conjunct evaluates to true if the q-bit AND/OR register is set to 'one', or its logical equivalent, and the r-bit OR conjunct/AND disjunct register is set to 'one', or its logical equivalent, and the processor short-circuits to the start of the next conjunct.

43. The processor of Claim 40, wherein a predetermined disjunct evaluates to false if the q-bit AND/OR register is set to 'zero', or its logical equivalent, and the r-bit OR conjunct/AND disjunct register is set to 'one', or its logical equivalent, and the processor short-circuits to the start of the next disjunct.

44. The processor of Claim 40, wherein the AND/OR registers and the OR conjunct/AND disjunct register are each 1 bit wide.

45. The processor of Claim 40, wherein the AND/OR registers and the OR conjunct/AND disjunct register are each more than one bit wide.

46. The processor of Claim 1, further comprising a control encoder, wherein the control encoder accepts n+m bits in parallel and outputs them across a device bus either in series or in parallel.

47. The processor of Claim 46, further comprising a control buffer, having one or more outputs connected to the control encoder, that is operable for storing state change commands until the control encoder is clear.

48. The processor of Claim 1, further comprising a secondary device state random-access memory, wherein the secondary device state random-access memory accepts n+m bits in parallel, stores the n+m bits in a memory location corresponding to a particular device of a plurality of devices, and, in response to a request from the particular device, outputs them to the device.

49. The processor of Claim 1, further comprising one or more memory devices, wherein the one or more memory devices are operable for storing the states of a plurality of devices that the processor monitors and controls.

50. The processor of Claim 49, wherein the states are each stored in a particular memory location, and wherein the random-access memory includes at least one bit associated with each memory location, and wherein the at least one bit is operable for indicating whether or not the device state stored in the respective memory location is in the process of being modified.

51. The processor of Claim 50, further comprising a circuit operable for setting the at least one bit of each memory location at the beginning of the process of modifying the device state

stored in the respective memory location and for clearing the at least one bit of the respective memory location at the end of the process of modifying the device state stored therein.

52. The processor of Claim 50, further comprising a circuit operable for preventing the processor from reading a device state from a memory location when the at least one bit associated with the memory location indicates that the device state stored in the memory location is in the process of being modified.

53. The processor of Claim 49, wherein the one or more memory devices are random-access memory devices.

54. The processor of Claim 1, further comprising a control store, wherein the control store is operable for holding a compiled micro-program.

55. The processor of Claim 54, further comprising a program counter, wherein the program counter is operable for fetching an instruction from the control store.

56. The processor of Claim 55, further comprising a device, wherein the device is operable for configuring the program counter for normal operation, unconditional jump operation, conditional jump operation, and Boolean short-circuit operation.

57. The processor of Claim 1, wherein the plurality of registers comprise a plurality of multi-bit registers.

58. The processor of Claim 57, wherein the plurality of multi-bit registers comprise an instruction register, a first address register and a second address register, wherein if the Boolean logic unit is operable for performing the short-circuit evaluation of Conjunctive Normal Form Boolean expressions/operations, then the first address register is a next operation address register and the second address register is an end of OR address register, wherein if the Boolean logic unit is operable for performing the short-circuit evaluation of Disjunctive Normal Form Boolean expressions/operations, then the first address register is an end of operation address register and

the second address register is an end of AND address register, and if the Boolean logic unit is operable for performing the short-circuit evaluation of both Conjunctive and Disjunctive Normal Form Boolean expressions/operations, then the first address register is a next operation/end of operation address register and the second address register is an end of OR/AND address register.

59. The processor of Claim 58, wherein the instruction register comprises a register that is n+m+x bits wide and includes an n-bit address, an m-bit control/state word, and an x-bit operational code.

60. The processor of Claim 59, wherein the instruction register comprises a register that is n+m+3 bits wide and includes an operational code that is 3 bits wide.

61. The processor of Claim 59, wherein the operational code is more than 3 bits wide.

62. The processor of Claim 59, wherein the first and second address registers are n+m bits wide, and wherein the Boolean logic unit is operable to load both the n-bit address and the m-bit wide control/state word from the instruction register into the first and second address registers to form an address that is n+m bits wide.

63. The processor of Claim 59, wherein the Boolean logic unit is operable to associate each of a plurality of logical addresses with a respective physical address in a control store, and operable to retrieve a particular physical address from the control store based on the associated logical address present in one of the first or second address registers.

64. The processor of Claim 58, wherein the first address register stores an address used for Boolean short-circuiting.

65. The processor of Claim 58, wherein the second address register stores the address of an instruction immediately following a conjunct comprising an OR clause, or the address of an instruction immediately following a disjunct comprising an AND clause.

66. A processing method, comprising:

starting an operation related to a Normal Form Boolean expression, wherein the Boolean expression comprises a conjunct or a disjunct;

evaluating the conjunct or disjunct; and

selectively short-circuiting a portion of the Boolean expression.

67. The processing method of Claim 66, wherein starting an operation includes starting an operation related to a Disjunctive Normal Form Boolean expression, wherein the Boolean expression comprises a disjunct, and wherein evaluating includes evaluating the disjunct.

68. The processing method of Claim 67, wherein the disjunct is a stand-alone term evaluated as an OR operation.

69. The processing method of Claim 67, wherein the disjunct comprises an AND clause.

70. The processing method of Claim 69, wherein each of a plurality of terms of the disjunct is evaluated as part of an AND operation.

71. The processing method of Claim 70, further comprising initializing the value of an AND-bit to a first predetermined value and setting the value of the AND-bit to a second predetermined value if a selected term of the plurality of terms evaluates to false, wherein the second predetermined value differs from the first predetermined value.

72. The processing method of Claim 71, wherein the first predetermined value is ‘one’, or its logical equivalent, and the second predetermined value is ‘zero’ or its logical equivalent.

73. The processing method of Claim 72, further comprising, in a disjunct comprising an AND clause, AND'ing the result of each AND operation with the current value of an AND register.

74. The processing method of Claim 73, further comprising, in the event that the AND register has a value of 'zero', or its logical equivalent, and an AND disjunct register is set to 'one', or its logical equivalent, evaluating the disjunct to false and short-circuiting to a next disjunct.

75. The processing method of Claim 74, further comprising joining an OR operation and the next disjunct and rolling the value of the AND register up to an OR register by OR'ing the value of the AND register with the value of the OR register.

76. The processing method of Claim 75, further comprising, in the event that the AND-bit has a value of 'true', or its logical equivalent, when the OR operation is processed, changing the OR-bit to a value of 'true' or its logical equivalent.

77. The processing method of Claim 76, further comprising setting the final value of the Boolean expression to 'true', or its logical equivalent, if the OR-bit has a value of 'true', or its logical equivalent, and short circuiting the remainder of the Boolean expression.

78. The processing method of Claim 66, wherein starting an operation includes identifying whether the Normal Form Boolean expression is a Conjunctive Normal Form Boolean expression comprising a conjunct or a Disjunctive Normal Form Boolean expression comprising a disjunct, wherein if the Normal Form Boolean expression is a Conjunctive Normal Form Boolean expression, evaluating includes evaluating the conjunct, and wherein if the Normal Form Boolean expression is a Disjunctive Normal Form Boolean expression, then evaluating includes evaluating the disjunct.

79. The processing method of Claim 66, wherein starting an operation includes starting an operation related to a Normal Form Boolean expression, wherein the Boolean expression comprises a plurality of conjuncts or a plurality of disjuncts, and wherein evaluating the conjunct or disjunct includes:

separating the Boolean expression into separate conjuncts or disjuncts; and

distributing each separate conjunct or disjunct to a separate Boolean processor for evaluation.

80. The processing method of Claim 79, wherein if the Boolean expression is a Conjunctive Normal Form Boolean expression, then selectively short-circuiting includes:

if a conjunct in a first separate Boolean processor results in a false evaluation, providing a signal, from the first separate Boolean processor to the other separate Boolean processors, that the entire expression is false.

81. The processing method of Claim 79, wherein if the Boolean expression is a Disjunctive Normal Form Boolean expression, then selectively short-circuiting includes:

if a disjunct in a first separate Boolean processor results in a true evaluation, providing a signal, from the first separate Boolean processor to the other separate Boolean processors, that the entire expression is true.

82. A computing device, comprising:

a general-purpose processor; and

a Boolean co-processor that accepts code, representative of Boolean code, from the general-purpose processor, the Boolean processor including:

a Boolean logic unit, wherein the Boolean logic unit is operable for performing the short-circuit evaluation of Conjunctive Normal Form Boolean expressions/operations, operable for performing the short-circuit evaluation of Disjunctive Normal Form Boolean expressions/operations, or operable for performing the short-circuit evaluation of both Conjunctive Normal Form Boolean expressions/operations and Disjunctive Normal Form Boolean expressions/ operations;

a plurality of input/output interfaces, wherein the plurality of input/output interfaces are operable for receiving a plurality of compiled Boolean expressions/operations and transmitting a plurality of compiled results; and

a plurality of registers.

83. The computing device of Claim 82, wherein the Boolean co-processor further comprises a p-bit first register, wherein the p-bit first register is operable for rolling up the results of conjuncts and/or disjuncts.

84. The computing device of Claim 83, wherein the default value of the p-bit first register is 'one', or its logical equivalent, when evaluating Conjunctive Normal Form Boolean expressions/operations and 'zero', or its logical equivalent, when evaluating Disjunctive Normal Form Boolean expressions/operations.

85. The computing device of Claim 83, wherein the p-bit first register initializes to a value of 'one', or its logical equivalent, after the start of a predetermined operational code when evaluating Conjunctive Normal Form Boolean expressions/operations and a value of 'zero', or its logical equivalent, when evaluating Disjunctive Normal Form Boolean expressions/operations.

86. The computing device of Claim 83, wherein the p-bit first register remains at a value of 'one', or its logical equivalent, if all of the conjuncts of a Conjunctive Normal Form Boolean expression/operation being evaluated are true, and wherein the p-bit register remains at a value of 'zero', or its logical equivalent, if all of the disjuncts of a Disjunctive Normal Form Boolean expression/operation being evaluated are false.

87. The computing device of Claim 83, wherein a Conjunctive Normal Form Boolean expression/operation is false if the p-bit first register is set to 'zero', or its logical equivalent, and the remainder of the Boolean expression/operation is short-circuited.

88. The computing device of Claim 83, wherein a Disjunctive Normal Form Boolean expression/operation is true if the p-bit first register is set to 'one', or its logical equivalent, and the remainder of the Boolean expression/operation is short-circuited.

89. The computing device of Claim 82, wherein the Boolean co-processor further comprises a q-bit second register, wherein the q-bit second register is operable for storing the results of the evaluation of one or more terms within conjuncts and/or disjuncts.

90. The computing device of Claim 89, wherein the Boolean co-processor further comprises an r-bit third register, wherein the r-bit third register indicates that the evaluation of a conjunct comprising an OR clause, or a disjunct comprising an AND clause, has begun.

91. The computing device of Claim 90, wherein the r-bit third register initializes to a value of 'zero', or its logical equivalent, when evaluating Conjunctive Normal Form Boolean expressions/operations and remains in that state until an OR expression/operation sets its value to 'one', or its logical equivalent, and initializes to a value of 'one', or its logical equivalent, when evaluating Disjunctive Normal Form Boolean expressions/operations and remains in that state until an AND expression/operation sets its value to 'one' or its logical equivalent.

92. The computing device of Claim 90, wherein a predetermined conjunct evaluates to true if the q-bit second register is set to 'one', or its logical equivalent, and the r-bit third register is set to 'one', or its logical equivalent, and the computing device short-circuits to the start of the next conjunct.

93. The computing device of Claim 90, wherein a predetermined disjunct evaluates to false if the q-bit second register is set to 'zero', or its logical equivalent, and the r-bit third register is set to 'one', or its logical equivalent, and the computing device short-circuits to the start of the next disjunct.

94. The computing device of Claim 82, wherein the Boolean co-processor further comprises an operation decoder, wherein the operation decoder is operable for deciphering an operational code and controlling units that are dependent upon the operational code.

95. The computing device of Claim 94, wherein functions of the operation decoder comprise

Boolean AND, Boolean OR, end of operation, no operation, unconditional jump, conditional jump, start of operation, and start of conjunct/disjunct.

96. The computing device of Claim 82, wherein the Boolean co-processor further comprises a control encoder, wherein the control encoder accepts n+m bits in parallel and outputs them across a device bus either in series or in parallel.

97. The computing device of Claim 82, wherein the Boolean co-processor further comprises one or more memory devices, wherein the one or more memory devices are operable for storing the states of a plurality of devices that the computing device monitors and controls.

98. The processor of Claim 97, wherein the one or more memory devices are random-access memory devices.

99. The computing device of Claim 82, wherein the Boolean co-processor further comprises a control store, wherein the control store is operable for holding a compiled micro-program.

100. The computing device of Claim 99, wherein the Boolean co-processor further comprises a program counter, wherein the program counter is operable for fetching an instruction from the control store.

101. The computing device of Claim 100, wherein the Boolean co-processor further comprises a device, wherein the device is operable for configuring the program counter for normal operation, unconditional jump operation, conditional jump operation, and Boolean short-circuit operation.

102. The computing device of Claim 82, wherein the plurality of registers comprise a plurality of multi-bit registers.

103. The computing device of Claim 102, wherein the plurality of multi-bit registers comprise an instruction register, a first address register and a second address register, wherein if the Boolean

logic unit is operable for performing the short-circuit evaluation of Conjunctive Normal Form Boolean expressions/operations, then the first address register is a next operation address register and the second address register is an end of OR address register, wherein if the Boolean logic unit is operable for performing the short-circuit evaluation of Disjunctive Normal Form Boolean expressions/operations, then the first address register is an end of operation address register and the second address register is an end of AND address register, and if the Boolean logic unit is operable for performing the short-circuit evaluation of both Conjunctive and Disjunctive Normal Form Boolean expressions/operations, then the first address register is a next operation/end of operation address register and the second address register is an end of OR/AND address register.

104. The computing device of Claim 103, wherein the instruction register comprises a register that is n+m+x bits wide and includes an n-bit address, an m-bit control/state word, and an x-bit operational code.

105. The computing device of Claim 103, wherein the first address register stores an address used for Boolean short-circuiting.

106. The computing device of Claim 103, wherein when evaluating Conjunctive Normal Form Boolean expressions/operations, the second address register stores the address of an instruction immediately following a conjunct comprising an OR clause, and when evaluating Disjunctive Normal Form Boolean expressions/operations, the second address register stores the address of an instruction immediately following a disjunct comprising an AND clause.

107. The computing device of Claim 106, wherein the instruction whose address is stored in the second address register is another conjunct or disjunct, respectively.

108. A hybrid processor, comprising:

a host processor, wherein the host processor is at least operable for performing comparison operations and register modifications; and

a Boolean processor core, comprising:

a Boolean short-circuit outcome calculation unit, wherein the Boolean short-circuit outcome calculation unit is operable for evaluating the short-circuit outcome of Conjunctive Normal Form Boolean expressions/operations, operable for evaluating the short-circuit outcome of Disjunctive Normal Form Boolean expressions/ operations, or operable for evaluating the short-circuit outcome of both Conjunctive Normal Form Boolean expressions/operations and Disjunctive Normal Form Boolean expressions/operations;

a plurality of input/output interfaces, wherein the plurality of input/output interfaces are operable for receiving, from the host processor, data related to a plurality of compiled Boolean expressions/operations and transmitting, to the host processor, data representative of the short-circuit outcome of a plurality of evaluated Normal Form Boolean expressions/operations; and

a plurality of registers.

109. The hybrid processor of Claim 108, wherein the Boolean processor core further comprises a p-bit first register, wherein the p-bit first register is operable for rolling up the results of conjuncts and/or disjuncts.

110. The hybrid processor of Claim 109, wherein the default value of the p-bit first register is 'one', or its logical equivalent, when evaluating Conjunctive Normal Form Boolean expressions/operations and 'zero', or its logical equivalent, when evaluating Disjunctive Normal Form Boolean expressions/operations.

111. The hybrid processor of Claim 109, wherein the p-bit first register initializes to a value of 'one', or its logical equivalent, upon activation of the Boolean processor core when evaluating Conjunctive Normal Form Boolean expressions/operations, and wherein the p-bit first register initializes to a value of 'zero', or its logical equivalent, upon activation of the Boolean processor core when evaluating Disjunctive Normal Form Boolean expressions/operations.

112. The hybrid processor of Claim 109, wherein the p-bit first register remains at a value of

‘one’, or its logical equivalent, if all of the conjuncts of a Conjunctive Normal Form Boolean expression/operation being evaluated are true, and remains at a value of ‘zero’, or its logical equivalent, if all of the disjuncts of a Disjunctive Normal Form Boolean expression/operation being evaluated are false.

113. The hybrid processor of Claim 109, wherein a Conjunctive Normal Form Boolean expression/operation is false if the p-bit first register is set to ‘zero’, or its logical equivalent, and the remainder of the Boolean expression/operation is short-circuited.

114. The hybrid processor of Claim 109, wherein a Disjunctive Normal Form Boolean expression/operation is true if the p-bit first register is set to ‘one’, or its logical equivalent, and the remainder of the Boolean expression/operation is short-circuited.

115. The hybrid processor of Claim 108, wherein the Boolean processor core further comprises at least one enable register, wherein at least one enable register is operable for permitting the results of the evaluation of one or more terms within conjuncts and/or disjuncts to be outputted.

116. The hybrid processor of Claim 115, wherein at least one enable register is operable for permitting the results of the final evaluation of conjuncts and/or disjuncts to be outputted.

117. The hybrid processor of Claim 115, wherein the Boolean processor core further comprises an r-bit third register, wherein the r-bit third register indicates that the evaluation of a conjunct comprising an OR clause, or a disjunct comprising an AND clause, has begun.

118. The hybrid processor of Claim 117, wherein the r-bit third register initializes to a value of ‘zero’, or its logical equivalent, when evaluating Conjunctive Normal Form Boolean expressions/operations and remains in that state until an address representative of a subsequent conjunct or instruction is provided by the host processor, and initializes to a value of ‘one’, or its logical equivalent, when evaluating Disjunctive Normal Form Boolean expressions/operations

and remains in that state until an address representative of a subsequent disjunct or instruction is provided by the host processor.

119. The hybrid processor of Claim 117, wherein a conjunct evaluates to true if a predetermined input is set to 'one', or its logical equivalent, and the r-bit third register is set to 'one', or its logical equivalent, and the host processor short-circuits to the start of the next conjunct.

120. The hybrid processor of Claim 117, wherein a disjunct evaluates to false if a predetermined input is set to 'zero', or its logical equivalent, and the r-bit third register is set to 'one', or its logical equivalent, and the host processor short-circuits to the start of the next disjunct.

121. The hybrid processor of Claim 108, wherein the host processor further comprises an operation decoder, wherein the operation decoder is operable for deciphering an operational code and controlling units in the Boolean processor core that are dependent upon the operational code.

122. The hybrid processor of Claim 108, further comprising one or more memory devices, wherein the one or more memory devices are operable for storing the states of a plurality of devices that the hybrid processor monitors and controls.

123. The processor of Claim 122, wherein the one or more memory devices are random-access memory devices.

124. The hybrid processor of Claim 108, further comprising a control store, wherein the control store is operable for holding a compiled micro-program.

125. The hybrid processor of Claim 124, wherein the host processor further comprises a program counter, wherein the program counter is operable for fetching an instruction from the control store.

126. The hybrid processor of Claim 108, wherein the plurality of registers comprise a plurality of multi-bit registers.

127. The hybrid processor of Claim 126, wherein the plurality of multi-bit registers comprise a first address register and a second address register, wherein if the Boolean short-circuit outcome calculation unit is operable for evaluating the short-circuit outcome of Conjunctive Normal Form Boolean expressions/operations, then the first address register is a next operation address register and the second address register is an end of OR address register, wherein if the Boolean short-circuit outcome calculation unit is operable for evaluating the short-circuit outcome of Disjunctive Normal Form Boolean expressions/operations, then the first address register is an end of operation address register and the second address register is an end of AND address register, and if the Boolean short-circuit outcome calculation unit is operable for evaluating the short-circuit outcome of both Conjunctive and Disjunctive Normal Form Boolean expressions/operations, then the first address register is a next operation/end of operation address register and the second address register is an end of OR/AND address register.

128. The hybrid processor of Claim 127, wherein the first register stores an address used by the host processor for Boolean short-circuiting.

129. The hybrid processor of Claim 127, wherein when evaluating Conjunctive Normal Form Boolean expressions/operations, the second address register stores the address of an instruction immediately following a conjunct comprising an OR clause, and when evaluating Disjunctive Normal Form Boolean expressions/operations, the second address register stores the address of an instruction immediately following a disjunct comprising an AND clause.

130. The hybrid processor of Claim 127, wherein at least one of the first and second address registers stores an address that is transmitted to the host processor by the plurality of input/output interfaces as the short-circuit outcome of a plurality of evaluated Normal Form Boolean expressions/operations.

131. The hybrid processor of Claim 130, wherein the address is transmitted to a memory device in the host processor.

132. The hybrid processor of Claim 131, wherein the address is transmitted to a register in the host processor.

133. The hybrid processor of Claim 131, wherein the address is transmitted to a program counter in the host processor.

134. The hybrid processor of Claim 127, wherein loading the first address register activates the Boolean processor core.

135. The hybrid processor of Claim 127, wherein a predetermined input from the host processor deactivates the Boolean processor core, thereby preventing the Boolean processor core from transmitting, to the host processor, any short-circuit outcome data.

136. The hybrid processor of Claim 127, wherein a predetermined input from the host processor resets an OR conjunct and/or AND disjunct register, thereby notifying the Boolean processor core than an OR conjunct or AND disjunct has ended.

137. The hybrid processor of Claim 108, wherein the host processor is a general-purpose processor.

138. The hybrid processor of Claim 108, further comprising a CNF/DNF register operable for controlling whether the Boolean processor core is operated in a Conjunctive Normal Form mode or a Disjunctive Normal Form mode on the basis of a predetermined input from the host processor.

139. A compiling method, comprising:

receiving a plurality of conditional tests;

based upon the received plurality of conditional tests, generating an operation, in computer-readable format, representative of a Boolean expression in Conjunctive Normal Form or Disjunctive Normal Form; and

storing the operation in a Boolean processor operable to evaluate the Boolean expression by processing the operation and to selectively short-circuit a portion of the Boolean expression.

140. The compiling method of Claim 139, wherein the generated operation includes a plurality of portions, wherein at least a first of the plurality of portions is more likely to create a short-circuit condition than at least a second of the plurality of portions, and wherein generating an operation includes ordering the plurality of portions within the operation such that the first of the plurality of portions is likely to be processed before the second of the plurality of portions.

141. The compiling method of Claim 140, further comprising:

determining the relative likelihood of at least the first and second of the plurality of portions to create a short-circuit condition.

142. The compiling method of Claim 141, wherein the determining and generating steps are repeated periodically.

143. The compiling method of Claim 141, wherein the determining step is carried out on the basis of the results of at least one short-circuited evaluation of the stored operation.

144. The compiling method of Claim 141, wherein determining includes determining a probability of the first of the plurality of portions to create a short-circuit condition and a probability of the second of the plurality of portions to create a short-circuit condition, the method further comprising:

storing, in a memory, the probability of the first of the plurality of portions to create a short-circuit condition and the probability of the second of the plurality of portions to create a short-circuit condition.

145. The compiling method of Claim 144, wherein the determining and storing steps are repeated after each evaluation of the operation.

146. The compiling method of Claim 140, wherein generating an operation includes considering whether the Boolean expression is in Conjunctive Normal Form or Disjunctive Normal Form.